Information encoding by shortened Reed-Solomon codes

The present invention concerns communication systems in which the data to be transmitted are subjected to a channel encoding in order to improve the fidelity of the transmission. It concerns more particularly decoding methods as well as the devices and apparatuses adapted to implement those methods.

It will be recalled that so-called "channel" encoding consists, when the "codewords" sent to the receiver are formed, of introducing a certain amount of redundancy in the data to be transmitted. More particularly, by means of each codeword, the information is transmitted that is initially contained in a predetermined number k of symbols taken from an "alphabet" of finite size q; on the basis of these k information symbols, calculation is made of a number $n > k$ of symbols belonging to that alphabet, which constitute the components of the codewords: $\underline{v} \equiv (v_0, v_1, ..., v_{n-1})$ (the symbol "≡" means "by definition"). The set of codewords obtained when each information symbol takes some value in the alphabet constitutes a sort of dictionary referred to as a "code" of "dimension" k and "length" n.

In particular, certain codes, termed "linear codes" are such that any linear combination of codewords (with the coefficients taken from the alphabet) is still a codeword. These codes may conveniently be associated with a matrix H of dimension $(n-k)\times n$, termed "parity matrix": a word $\underline{v}$ of given length n is a codeword if, and only if, it satisfies the relationship: $H \cdot \underline{v}^T = 0$ (where the exponent T indicates the transposition); the code is then said to be "orthogonal" to the matrix H.

Among these linear codes, certain have the property of being "cyclic": for any given codeword $(v_0, v_1, ..., v_{n-1})$, the word $(v_{n-1}, v_0, v_1, ..., v_{n-2})$ obtained by cyclic permutation is a word of the same code. The words belonging to a cyclic linear code may conveniently be represented by means of polynomials: if the polynomial $v(x) \equiv \sum_{i=0}^{n-1} v_i x^i$ is made to correspond to some word $\underline{v} = (v_0, v_1, ..., v_{n-1})$, a given word $\underline{v}$ belongs to a given cyclic linear code if,

and only if, the corresponding polynomial $v(x)$ is a multiple of a certain polynomial $g(x)$, of degree $(n-k)$ and divisor of (x^n-1), termed "generator polynomial" of the cyclic linear code which characterizes it. Moreover a parity matrix *H* can be associated with any generator polynomial $g(x)$.

At the receiver, the decoding method correlative to a given encoding method judiciously exploits the redundancy incorporated in the codewords to detect any transmission errors and if possible to correct them. There is a transmission error if the difference $\underline{e}$ between a received word $\underline{r}$ and the corresponding codeword $\underline{v}$ sent by the transmitter is non-zero.

More particularly, the decoding is carried out in two main steps.

The first step consists of associating an "associated codeword" with the received word. Conventionally, the decoder first of all calculates the vector of "error syndromes" $H \cdot \underline{r}^{\mathrm{T}} = H \cdot \underline{e}^{\mathrm{T}}$. If the syndromes are all zero, it is assumed that no transmission error has occurred, and the "associated code word" will then simply be taken to be equal to the received word. If that is not the case, it is thereby deduced that certain symbols in the received word are erroneous, and a correction algorithm is then implemented which is adapted to estimate the value of the error $\underline{e}$; the algorithm will thus provide an estimated value $\hat{\underline{e}}$ such that $(r-\hat{\underline{e}})$ is a codeword, which will then constitute the "associated codeword".

The second step simply consists in reversing the encoding method. In the ideal situation in which all the transmission errors have been corrected, the initial information symbols are thereby recovered.

The purpose of an error correction algorithm is to associate with the received word the codeword situated at the shortest Hamming distance from that received word, the "Hamming distance" being, by definition, the number of places where two words of the same length have a different symbol. The shortest Hamming distance between two different codewords of a code is termed the "minimum distance" *d* of that code. This is an important parameter of the code. More particularly, it is in principle possible to find the position of the possible errors in a received word, and to provide the correct replacement symbol (i.e. that is identical to that sent by the transmitter) for each of those

positions, each time the number of erroneous positions is at most equal to INT[(*d*–1)/2] (where "INT" designates the integer part) for a code of minimum distance *d* (for certain error configurations, it is sometimes even possible to achieve better). However, in all cases, the concern is not with a possibility in principle, since it is often difficult to develop a decoding algorithm achieving such performance.

The present invention concerns in particular "Reed-Solomon" codes, which are reputed for their efficiency. These are cyclic linear codes, of which the minimum distance *d* is equal to $(n-k+1)$. In general terms, codes of dimension k and of length n having a minimum distance $d = n-k+1$ are termed "MDS" (for "*Maximum Distance Separable*") since they are codes of which the minimum Distance allows a Maximum Separation between the codewords, account taken of the parameters k and n).

When the size q of the "alphabet" is a power of a prime number, the alphabet can be given the structure of what is known as a "Galois field" denoted $\mathbf{F}_q$, of which the non-zero elements may conveniently be identified as each being equal to γ^{i-1} for a corresponding value of *i*, where $i = 1,\ldots,q-1$, and where γ is a primitive $(q-1)^{\text{th}}$ root of unity in $\mathbf{F}_q$.

The parity matrix H of the Reed-Solomon code of dimension k and of length n (where n is necessarily equal to $(q-1)$ or a divisor of $(q-1)$) is a matrix with $m \equiv n-k$ lines and n columns, which may be defined for example by taking $H_{ij} = \alpha^{(i+1)j}$ $(0 \le i \le m-1,\ 0 \le j \le n-1)$, where α is an n^{th} root of unity in $\mathbf{F}_q$. The generator polynomial of this code is $g(x) = \prod_{i=1}^{m}(x-\alpha^i)$.

Among the algorithms known for encoding a sequence $\underline{a} = (a_0, a_1, \ldots, a_{k-1})$ of information symbols belonging to $\mathbf{F}_q$ by means of a Reed-Solomon code, certain use that parity matrix *H*. In these algorithms, a certain relationship is chosen between the information symbols and those of the corresponding codeword $\underline{v}$ (for example $v_i = a_i$ for $0 \le i \le k-1$; in this case, the encoding is said to be "systematic"). Next the components of $\underline{v}$ remaining to

$$v^*(x) = a(x) - r(x) \equiv \sum_{i=0}^{n'-1} v^*_i \, x^i \ ,$$

corresponding to the word $\underline{v}^* = (v^*_0, v^*_1, \ldots, v^*_{n'-1})$ belonging to C'. It should be noted that, by construction, the contribution of $a(x)$ to $v^*(x)$ only contains powers of x belonging to $\underline{s}$ (the smallest being greater than or equal to s_m).

The two main cases are then to be envisaged for the fourth step of the method according to the invention.

If the m first integers of $\underline{s}$ comprise all the successive integers from 0 to $(m-1)$, then the contribution of $r(x)$ to $v^*(x)$ only contains powers of x belonging to $\underline{s}$ (the largest being less than or equal to $s_{m-1} = m-1$). In this case, it suffices to take $\underline{v}' = \underline{v}^*$.

On the other hand, if there is at least one "hole" in the succession of the m first integers of $\underline{s}$ (and thus $s_{m-1} > m-1$, $s_m > m$, and so forth), calculation will now be made of a word $\underline{v}'$ satisfying:

- $v'_i = v^*_i$ for $s_m \le i \le n'-1$ (contribution of $a(x)$), and
- $v'_i = c_i$, in which the c_i are predetermined constants, for all $i < s_m$ not belonging to $\underline{s}$, that is to say different from s_0, s_1, ..., s_{m-1}; these values of i are thus $(s_m - m)$ in number.

It will be noted that $v^*_i = 0$ for $m \le i \le (s_m - 1)$; but, in the case considered in which $s_{m-1} > m-1$, $v'_i = 0$ will not necessarily be satisfied for those indices, even when $c_i = 0$. This is because, according to the invention, a word $\underline{v}'$ belonging to C' is obtained by linear combination of $\underline{v}^*$ with e, where $e \equiv s_m - m$, judiciously chosen words $\underline{\Gamma}^j$ belonging to C':

$$\underline{v}' = \underline{v}^* + \sum_{j=0}^{e-1} f_j \underline{\Gamma}^j \ , \qquad (1)$$

in which the coefficients f_j belong to $\mathbf{F}_q$. Explicitly, this means:

- for $s_m \le i \le n'-1$: $v'_i = v^*_i$, (1a)

- for $m \le i < s_m$: $v'_i = \sum_{j=0}^{s_m - m - 1} f_j \Gamma^j{}_i$, and (1b)

- for $0 \le i < m$: $v'_i = v^*_i + \sum_{j=0}^{s_m - m - 1} f_j \Gamma^j{}_i$. (1c)

The words $\underline{\Gamma}^j$ are constructed from the coefficients of $g(x) \equiv \sum_{p=0}^{m-1} g_p x^p$. More particularly, the words $\underline{\Gamma}^j$ are defined as follows:

for $j \le i \le j + m$: $\Gamma^j{}_i = g_{i-j}$, and (3a)

- $\Gamma^j{}_i = 0$ otherwise. (3b)

Explicitly:

$\underline{\Gamma}^0 = (g_0, g_1, \ldots, g_m, 0, \ldots, 0)$,

$\underline{\Gamma}^1 = (0, g_0, g_1, \ldots, g_m, 0, \ldots, 0)$,

...

$\underline{\Gamma}^{e-1} = (0, \ldots, 0, g_0, g_1, \ldots, g_m, 0, \ldots, 0)$, in which the index of the last non-zero component is $i = s_m - 1$.

The polynomial corresponding to the word $\underline{\Gamma}^0$ being identical to $g(x)$, it is of course divisible by $g(x)$, and $\underline{\Gamma}^0$ thus belongs to C'. The other words $\underline{\Gamma}^j$ just being circular permutations of $\underline{\Gamma}^0$, they also belong to C'.

It then only remains to find the e coefficients of f_j which satisfy the system of e equations concerned (in which it is stipulated that $v'_i = c_i$, as mentioned above) among the n' linear equations represented by equation (1). It can be shown that, with the definitions (3a-3b), the determinant of that system is non-zero whatever said equations concerned are; this property is linked to the "MDS" character of the code C'.

The method described above will now be illustrated by means of a numerical example.

Take, for example, for code C', the one which served above by way of example, to illustrate the Euclidean algorithm ($q = 2^4$, $k' = 10$, $n' = q - 1 = 15$, $m = 5$, and $g(x)$ given by equation (2)). It is desired to shorten it to construct a code C of dimension $k = 5$ and length $n = 10$; more specifically, suppose that it is wished to obtain $v'_i = 0$ for i not belonging to

$$\underline{s} = (1,2,4,5,7,8,10,11,13,14) .$$

Thus, $s_m = 8$, and $e = 8 - 5 = 3$, and the three indices in question (that is to say not belonging to $\underline{s}$, and less than $s_m = 8$) are: 0, 3 and 6. This then gives:

$$\underline{\Gamma}^0 = (1, \gamma, \gamma^5, \gamma^2, \gamma^7, 1, 0, \ldots, 0) ,$$

$$\underline{\Gamma}^1 = (0, 1, \gamma, \gamma^5, \gamma^2, \gamma^7, 1, 0, \ldots, 0) ,$$

$$\underline{\Gamma}^2 = (0, 0, 1, \gamma, \gamma^5, \gamma^2, \gamma^7, 1, 0, \ldots, 0) ,$$

and thus, in particular:

$$\Gamma^0{}_0 = 0 , \ \Gamma^0{}_3 = \gamma^2 , \ \Gamma^0{}_6 = 0 ,$$

$$\Gamma^1{}_0 = 0 , \ \Gamma^1{}_3 = \gamma^5 , \ \Gamma^1{}_6 = 1 , \text{ and}$$

$$\Gamma^2{}_0 = 0 , \ \Gamma^2{}_3 = \gamma , \ \Gamma^2{}_6 = \gamma^7 .$$

Let for example the following information sequence be encoded

$$\underline{a} = (\gamma^9, 0, \gamma^{11}, 0, 0) .$$

The following polynomial is then made to correspond to it

$$a(x) = \gamma^{11} x^{11} + \gamma^9 x^8 ,$$

and Euclidean division by $g(x)$ gives:

$$q(x) = \gamma^{11} x^6 + \gamma^3 x^5 + \gamma^9 x^4 + \gamma^6 x^3 + \gamma^{14} x^2 + \gamma^{13} + \gamma^8 , \text{ and}$$

$$r(x) = \gamma x^4 + \gamma x^3 + \gamma^{13} x^2 + \gamma^{10} x + \gamma^8 .$$

Consequently,

$$v^*(x) = \gamma^{11} x^{11} + \gamma^9 x^8 + \gamma x^4 + \gamma x^3 + \gamma^{13} x^2 + \gamma^{10} x + \gamma^8 ,$$

and thus: $v^*{}_0 = \gamma^8$, $v^*{}_3 = \gamma$ (naturally $v^*{}_6 = 0$).

Solving the 3 equations (1) then gives:

$f_0 = \gamma^8,\ f_1 = \gamma^2,\ f_2 = \gamma^{10}$.

Finally, the following is obtained:

$$\underline{v}' = \underline{v}^* + \gamma^8 \underline{\Gamma}^0 + \gamma^2 \underline{\Gamma}^1 + \gamma^{10} \underline{\Gamma}^2 = \left(0, \gamma^{14}, \gamma^{12}, 0, 1, 0, 0, \gamma^{10}, \gamma^9, 0, 0, \gamma^{11}, 0, 0, 0\right).$$

In this embodiment, once the encoding is terminated, the encoding unit 102 transmits the "pre-encoded" words $\underline{v}'$ to a shortening unit 20, which deletes the components of $\underline{v}'$ of which the index does not belong to the set $\underline{s}$. Thus the words $\underline{v}$ belonging to the shortened code C are obtained.

Thus, in the numerical example which has just been considered, the following is obtained:

$$\underline{v} = \left(\gamma^{14}, \gamma^{12}, 1, 0, \gamma^{10}, \gamma^9, 0, \gamma^{11}, 0, 0\right).$$

It is clear, in view of the above account, that the lower the value of e, the lesser will be the calculations implied by the method according to the invention. It will now be shown how it is possible to minimize the value of e according to a refinement to the invention, for given $\underline{s}$.

As was shown above, the value of e is linked to the existence of "holes" in a succession of m consecutive elements of $\underline{s}$. The refinement in question thus consists of searching in $\underline{s}$ for the sequence of m consecutive elements having the least "holes", and of bringing that sequence to the beginning of the words of the code C', taking advantage of the cyclic character of that code.

This strategy will be better understood with the aid of a numerical example. Let us resume the example in which the polynomial generator is given by equation (2), but this time a set of predetermined positions will be considered which is given by:

$$\underline{s} = (1, 4, 6, 7, 8, 9, 11, 12, 13, 14)\ .$$

Here $s_m = s_5 = 9$, and thus the strict application of the method described above leads to forming the following polynomial from an information sequence $\underline{a}$ of length 5:

$$a(x) = a_0 x^9 + a_1 x^{11} + a_2 x^{12} + a_3 x^{13} + a_4 x^{14}\ .$$

Euclidean division makes it possible to construct the word

$$\underline{v}^* = (-r_0, -r_1, -r_2, -r_3, -r_4, 0,0,0,0, a_0, 0, a_1, a_2, a_3, a_4) ,$$

in which it is proposed, for example, to transform the $e = 4$ components v^*_0, v^*_2, v^*_3, and v^*_5 into constants which are all zero in the manner taught by the invention.

It will now be shown that there is another manner of performing the calculations, which is appreciably less complex since it leads to a lower value of e, i.e. $e^* = 1$.

Indeed, it is noted that, in $\underline{s}$, the sequence of 5 consecutive positions

8,9,11,12,13

has only a single "hole", i.e. position No. 10. Thus, if on all the words of code C', a circular permutation of 8 positions to the left is performed, the initial position of that sequence, i.e. position No. 8, is brought to initial position No. 0. Overall, $\underline{s}$ is thus transformed into

$$\underline{s}^* = (0,1,3,4,5,6,8,11,13,14) ,$$

recalling that the positions of components are defined modulo $n' = 15$. Thus, in $\underline{s}^*$, the sole "missing" position below $s^*_m = s^*_5 = 6$ is position No. 2, which corresponds indeed to $e^* = 1$.

In polynomial "language", this circular permutation of 8 positions to the left corresponds to a multiplication by x^{-8} modulo $(x^{15} - 1)$. Now make the following polynomial correspond to the information sequence $\underline{a}$:

$$a(x) = a_0 x + a_1 x^4 + a_2 x^6 + a_3 x^7 + a_4 x^{14} ,$$

and define

$$a^*(x) \equiv x^{-8} a(x) = a_4 x^6 + a_0 x^8 + a_1 x^{11} + a_2 x^{13} + a_3 x^{14} .$$

The Euclidean division of $a^*(x)$ by $g(x)$ then gives a remainder

$$r^*(x) = r^*_4 x^4 + r^*_3 x^3 + r^*_2 x^2 + r^*_1 x + r^*_0 .$$

It is thereby deduced that $v^*(x) \equiv a^*(x) - r^*(x)$, hence

$$\underline{v}^* = (-r^*_0, -r^*_1, -r^*_2, -r^*_3, -r^*_4, 0, a_4, 0, a_0, 0, 0, a_1, 0, a_2, a_3) .$$

In this word $\underline{v}^*$, solely the component $v^*_2 = -r^*_2$ is both non-zero (in general) and absent from $\underline{s}^*$. The technique taught according to the

invention is next implemented, now with $e^* = 1$, to deduce from $\underline{v}^*$ a codeword $\underline{v}'$ having $v'_2 = 0$.

It remains nevertheless to apply a circular permutation of 8 positions to the components of this word $\underline{v}'$, this time to the right, to get back to the positions prescribed by $\underline{s}$. In this way a word of C' is obtained which is the final result of the encoding according to the invention of the information sequence $\underline{a}$.

The words $\underline{v}$ issuing from the shortening unit 20 are finally transmitted by the transmission unit 103 to a predetermined recipient. This recipient may for example form part of a complex encoding system (for example relying on a multiplicity of shortened Reed-Solomon codes). This recipient may, according to another example, be a transmission chain comprising a modulator, which associates a modulation symbol with each predetermined number of binary symbols (bits), followed by a recorder or else (respectively) by a transmitter inserting the symbols in a transmission channel, that channel for example able to be a storage on a suitable carrier (such as a DVD, or a magnetic or magnetico-optical disc, or else magnetic tape), or (respectively) transmission by wire or wireless transmission (such as a radio link).

The block diagram of **Figure 2** represents, very schematically, a data processing apparatus 48 incorporating the encoder 102.

This apparatus 48 comprises a keyboard 911, a screen 909, a source of external information 100, a transmitter 103, conjointly connected to input/output ports 903 of an encoding device 102 which is implemented here in the form of a logic unit.

The encoding device 102 comprises, connected together by an address and data bus 902:

- a central processing unit 900,
- a random access memory RAM 904,
- a read only memory 905, and
- said input/output ports 903.

Each of the elements illustrated in Figure 2 is well known to a person skilled in the art of microcomputers and transmission systems and, more

generally, of information processing systems. These known elements are therefore not described here. It should be noted, however, that:

- the information source 100 could, for example, be an interface peripheral, a sensor, a demodulator, an external memory or other information processing system (not shown), and could for example supply sequences of signals representing speech, service messages or multimedia data in particular of IP or ATM type, in the form of sequences of binary data, and

- the transmitter 103 is adapted to transmit the words belonging to code C, for example to a unit belonging to a complex encoding system, or to a device for sending on a radio channel or for recording on a carrier for mass storage.

The random access memory 904 stores data, variables and intermediate processing results, in memory registers bearing, in the description, the same names as the data whose values they store. It should be noted, in passing, that the word "register" designates, throughout the present description, a memory area of low capacity (a few items of binary data) and equally well a memory area of high capacity (for storing a complete program) within a random access memory or read only memory.

The random access memory 904 contains in particular the following registers:

- a register *"information_symbols"* in which the information symbols belonging to $\mathbf{F}_q$ are stored,

- a register "*pre-encoded_words*", in which the words $\underline{v}'$ belonging to the non-shortened code are stored, and

- a register "*code_words*", in which the words $\underline{v}$ belonging to the shortened code are stored, before being submitted to the transmitter 103.

The read only memory 905 is adapted to store, in registers which, for convenience, have the same names as the data which they store:

- the operating program of the central processing unit 900, in a register "program",

- the length n' of the words belonging to the non-shortened code, in a register "n'",

- the length n of the words belonging to the shortened code, in a register "n",

- the set $\underline{s}$ of the positions of the components to keep after shortening, in a register "s",

- the cardinal of the Galois field $\mathbf{F}_q$ serving as alphabet for the code used, in a register "q",

- the number $k = n - m$ of information symbols serving to construct a codeword, in a register "k", and

- the coefficients of the polynomial generator $g(x)$ of the non-shortened code, in a register "g".